Why MLLMs Struggle to Determine Object Orientations

Anju Gopinath Nikhil Krishnaswamy Bruce Draper
Department of Computer Science
Colorado State University
Fort Collins, CO, USA
anju@colostate.edu

Abstract

Multimodal Large Language Models (MLLMs) struggle with tasks that require reasoning about 2D object orientation in images, as documented in prior work. Tong et al. and Nichols et al. hypothesize that these failures originate in the visual encoder, since commonly used encoders such as CLIP and SigLIP are trained for image–text semantic alignment rather than geometric reasoning. We design a controlled empirical protocol to test this claim by measuring whether rotations can be recovered from encoder representations. In particular, we examine SigLIP and ViT features from LLaVA-OneVision and Qwen2.5-VL-7B-Instruct models, respectively, using full images, and examine CLIP representations in LLaVA-1.5 and 1.6 using rotated foreground patches against natural background images. Our null hypothesis is that orientation information is not preserved in the encoder embeddings and we test this by training linear regressors to predict object orientation from encoded features. Contrary to the hypothesis, we find that orientation information is recoverable from encoder representations: simple linear models accurately predict object orientations from embeddings. This contradicts the assumption that MLLM orientation failures originate in the visual encoder.

Having rejected the accepted hypothesis that MLLMs struggle with 2D orientation tasks because of visual encoder limitations, we still don't know why they fail. Although a full explanation is beyond the scope of this paper, we show that although present, orientation information is spread diffusely across tens of thousands of features. This may or may not be while MLLMs fail to exploit the available orientation information.

1. Introduction

Multimodal Large Language Models (MLLMs) struggle with tasks that require identifying orientations of objects in images [4, 6, 12, 16, 21, 24, 25, 32–35, 37, 44, 45]. For example, Nichols et al. tested 15 MLLMs on questions requiring granular knowledge of object orientations and found that the best systems could only answer 34% of the questions correctly [24]. Similarly, Kamoi et al. evaluated 23 MLLMs on visual question answering (VQA) tasks that require knowledge of shapes and angles, and found that the best were acccurate only 60% of the time [14]. On the Visual-Spatial Intelligence Benchmark (VSI-Bench), Yang et al. discovered that MLLMs performed near or below chance-level performance and [45] report the highest score of 46.3 for a proprietary model and 43.3 for an open source model (7B parameters) vs. 39.5 (78B parameters) for relative directional tasks among 11 MLLMs [35, 45] . This observation is corroborated by Zhang et al.'s finding that increasing the training data alone is not sufficient to improve performance across all spatial tasks [42]. Even worse, linguistic reasoning techniques such as *Chain-of-Thought (CoT), Self-Consistency* and *Tree-of-Thoughts* led to a significant performance degradation on the benchmark, highlighting the lack of a robust spatial reasoning module in the architecture of MLLMs [35].

The reason why MLLMs fail at 2D orientation tasks is unknown, but Tong et al. [32] and Nichols et al. [24] suggest that the initial visual encoders may be at fault. Most visual encoders are not trained on orientation tasks, and if object orientation information is not embedded in the visual encoding, there is no way for an MLLM to recover it.

This paper empirically tests the hypothesis that MLLMs struggle with orientation tasks because their visual encoders fail to embed orientation information. In particular, we measure how well a linear regressor can predict the orientation of an object or image from its visual embedding. Since this requires access to a network's internal representations, we test four open-source systems (LLaVA-OneVision [15], Qwen2.5-VL-7B [1], LLaVA-v1.5-13B [18], and LLaVA-v1.6-vicuna-13B [19]) which between them use three open-source visual encoders (ViT [7], SigLIP [40], and CLIP [27]). For systems that take two images as input, we give them an image and a rotated version of the same image and

ask for the degree of rotation, while for single image input systems, we rotate a foreground object and ask for the rotation of the foreground object relative to the (unrotated) background. Not surprisingly, all four systems fail at their assignments – we have already noted that MLLMs struggle with orientation tasks. What we measure is how well linear regressors can predict the rotations from the internal visual embeddings.

Our key finding is that linear regressors *can* predict image and object rotations to within $\pm 3^\circ$ from the visual embeddings. This contradicts the original hypothesis and raises a new question: if the fault is not with the visual encoders, why do MLLMs struggle with orientation tasks? We do not conclusively answer this question in its entirety, but we explore some properties of rotation information in visual embeddings. We discover (1) that errors in rotation estimation are not only small but approximately Gaussian and (2) that orientation information is diffusely distributed across tens of thousands of values in the embedding vector.

To summarize, our contributions are:

(1) We show that the SigLIP, ViT and CLIP vision encoders capture the orientations of images and foreground objects with a high degree of accuracy (MAE $< 3^\circ$).

(2) We show that the errors in orientation predictions based on SigLIP, ViT and CLIP encoding are roughly Gaussian and distributed across tens of thousands of features.

For anyone wishing to replicate this work, the images and code can be found in https://github.com/anjugopinath/MLLM_Orientation.

2. Related Work

2.1. Orientation Estimation Capabilities of Multimodal Large Language Models

Existing works have identified the gap in visual-spatial intelligence of MLLMs and have attributed it to their limited spatial reasoning capabilities [3, 4, 6, 9, 12, 21, 26, 35, 37, 38, 43, 44]. Orientation (or relative direction) estimation is one of the metrics in these works that performs poorly.

Previous works addressing object orientation have constructed benchmarking datasets containing coarse-grained questions (e.g. left vs. right, above vs. below, and front vs. behind) [4, 13, 29] that test the understanding of relative orientation and fine-grained questions (0°, 90°, 180°, 270°) [25] that test the accuracy in estimating a few canonical orientations. Jung et al. argue that inconsistent data annotation leads to poor results on object orientation estimation tasks [13]. Nichols et al. [24] and Tong et al. [32] hypothesize that the reason why MLLMs perform well on categorical clustering of directions into "left" and "right" but fail at angular estimation is because they are pretrained on CLIP-like models which focus on image-text semantic alignment rather than on geometric understanding. Similary, Yoon et al. conclude that MLLMs lose fine-grained visual information due to the visual instruction tuning paradigm [36], as does Liu et al. [17]. Huynh et al. analyzes orientation estimation capabilities of LVLMs using odd-one-out experiments and concludes that the poor performance is due to the information provided by the vision encoder not being discriminative enough [11]. We note that these works do not analyze the performance of the vision encoder in MLLMs in isolation on orientation estimation. In contrast, we perform a fine-grained analysis for every 1° orientation between $0 - 360^\circ$ and conclude that even though the language models perform poorly the vision encoders embed orientations to a high degree of accuracy.

2.2. Feature/Latent Space Manipulation

For both LLMs and vision-only models, there exist methods for interpretability and feature analysis. In vision-only models, disentangling CNN representations have been studied to understand how a visual pattern might describe object parts or textures [41]. In LLMs, methods such as Representation Control [2] involves editing the latent space of a neural network to steer the model towards a particular output by injecting a vector in between the layers of a model at inference. While these methods involve architectural changes to analyze the interpretability of Large Language Models, analyzing and controlling representations in vision-language models remains challenging [31]. Cui et al. performs interchange intervention to analyze ordering information contained in VLMs [5]. Faced with the unique challenge of identifying how vision-language models such as LLaVA-LLaMA and Qwen2.5-VL-7B-Instruct might be encoding orientation awareness in their vision encoders, we analyze the visual embeddings and perform feature substitution by iteratively substituting features from an anchor embedding into a target embedding with the goal of identifying the features that encode orientation.

2.3. Orientation Estimation by Neural Networks

There is an older line of research asking whether Convolutional Neural Nets (CNNs) can tell if an image has been rotated from its original orientation. Sun et al. [30] perform orientation estimation using image features on the Outdoor Images dataset collected from the Flickr1M dataset. The orientations were manually annotated. The work titled OSKDet performs orientation-sensitive keypoint based rotated object detector [20]. Fischer et al. perform exact orientation estimation on general natural images. Images from the Microsoft COCO dataset were artificially rotated and in the test set, slanted images, framed images and images which do not have a well-defined orientation were discarded [8]. Their method estimates orientation with an average accuracy of 3° in the setting with $\pm 30^\circ$. Their model was

built on the AlexNet architecture pretrained on ImageNet. While existing works on orientation estimation that utilize features from image-based feature extractors are predominantly vision-only models, in this work, we study orientation estimation using features from LLaVA's vision encoder.

3. Methodology

Image Set	Prompt
Whole Images	How much is the 2nd image rotated clockwise when compared to the first image?
In-Place Rotated Images	How much is the <object(s)> rotated clockwise in degrees when compared to the first image?
dog on beach	By how much is the dog in the center inside the circle rotated if it is known that the rotation is zero degrees when the dog's legs are vertical?
lizard on fish	By how much is the lizard in the center inside the circle rotated if it is known that the rotation is zero degrees when the lizard is roughly horizontal with its tail pointing left and its head pointing right?
train on indoor	By how much is the train in the center inside the circle rotated if it is known that the rotation is zero degrees when the train is vertical with the train tracks at the bottom and the steam from the train going vertically upwards?

Table 1. Image sets and corresponding prompts to LLaVA/Qwen2.5-VL-7B-Instruct.

We test the hypothesis suggested by Nichols et al. that MLLMs struggle with orientation queries because CLIP-style encoders do not preserve object orientation information [24]. It is an appealing hypothesis because CLIP was trained for semantic contrastive alignment, not orientation tasks, but we find the hypothesis is not true. To test it, we use natural images for LLaVA-OneVision and Qwen2.5-VL-7B-Instruct (2 image version) and superimpose circular patches of foreground images onto natural background images for LLaVA1.5 and 1.6 (single image version). The superimposed images look odd (see Figure 1, section C), but they allow us to apply arbitrary rotations to the foreground without introducing rotation-induced artifacts at patch boundaries. We present these images with a text query to LLaVA-LLaMA models (OneVision, 1.5 and 1.6) and Qwen2.5-VL-7B-Instruct and extract the embedding vectors produced by the visual encoder (SigLIP for LLaVA-OneVision, ViT for Qwen and CLIP for LLaVA 1.5 and 1.6). We then train a linear ridge regressor to predict orientations of the image (2 image version - LLaVA-OneVision and Qwen2.5-VL-7B-Instruct) or foreground orientations given inputs composed of a single background image with a foreground patch superimposed at different orientations (single image version - LLaVA 1.5 and 1.6), and test it using novel whole image or foreground patch orientations respectively. The objective is to test whether the orientation of the whole image or of the foreground patch is encoded in the visual embedding vector when all other factors are held constant.

Probing further, we estimate the accuracy of the orientation predictions for 6 image sets (2 image version - LLaVA-OneVision and Qwen2.5-VL-7B-Instruct) and at different foreground scales and with three different foreground/background pairs (single image version - LLaVA 1.5 and 1.6). We find that the mean average prediction error is always less than 3° and that the errors are roughly Gaussian. This implies that LLaVA's and Qwen2.5-VL-7B-Instruct's visual encoder does preserve orientation information, and that the orientation signal is accurate and well-behaved. The rest of this section describes the experiment and its analysis in more detail.

3.1. Constructing Image Samples

We construct 3 distinct types of image sets for experiments as detailed in the subsections below. Models that accept multiple image inputs are prompted to determine the orientation of a rotated image compared to an unrotated image, while models that take only a single image input are asked for the orientation of a rotated foreground patch relative to an unrotated background image.

3.1.1. Cropped Whole Images and Images with In-Place Rotated Objects for LLaVA-OneVision and Qwen2.5-VL-7B

We perform experiments with two categories of images - whole images where the whole image is rotated and images with in-place rotated objects where selected objects are rotated keeping the background static. For the first category, we selected six images from ImageNet [28]. Each image was rotated in 1° increments, and the central region was cropped to remove blank canvas artifacts introduced by rotation, which could otherwise affect regression accuracy. After preprocessing, each image set contained 180 samples. For the second category, we utilize the SI-Score dataset [39] and code, and generate 3 image sets, each with 180 samples. Two of them - *koala* and *vase* are instances of a single object rotated in-place while *vase and toaster* is an instance of two objects rotated in-place. The resultant images from both categories are either 200 by 200 or 250 by 250 in size. The first image from each set from the two categories are shown in Figure 1, sections A and B.

3.1.2. Blended Image Samples for LLaVa1.5 and 1.6

Newer MLLMs that take multiple image inputs do not use the older CLIP encoder. To test whether the CLIP encoder preserves foreground orientation information, we created test images where the 2D orientation of the foreground could be carefully controlled. We started with three natural images from ImageNet [28] for the background: a beach scene, a top-view of a koi pond, and an indoor picture of a kitchen. Next, the rectangular source foreground

Figure 1. Set of images - Sets A and B are used for experiments with LLaVA-OV and Qwen2.5-VL-7B-Instruct, and set C is used for LLaVA1.5 and LLaVA1.6. Since LLaVA-OV and Qwen2.5-VL-7B-Instruct can be prompted with 2 images, real images were used. But since LLaVA1.5 and 1.6 can be prompted with only 1 image, synthetic images with reduced artifacts were used to improve the prediction accuracy.
[A: Whole Images]: (a) dog (b) train (c) lizard (d) beach (e) indoor (f) fish.
[B: In-Place Rotated Objects]: (a) koala on beach (b) vase on indoor (c) vase and toaster on indoor.
[C: Blended Images with foreground on background]: (a) dog on beach (b) lizard on fish (c) train on indoor environment, with foreground scales 1, 2 and 3 from left to right.

image (also from ImageNet) is padded using black pixels to obtain a square. A black square array having the same dimensions as the padded source is created on which a circle is drawn using alpha blending with a mask value of 1.0 (opaque) inside the circle which gradually fades to 0.0 (transparent) outside it ($Blended = Foreground \times Mask + Background \times (1 - Mask)$). This ensures a smooth transition, avoiding jagged boundaries. This process is repeated every time as the foreground patch is rotated in 1° increments. The final output is a rectangular background (bg) image with a circular patch (fg) in the center which serves as the foreground object. We used three sizes (in pixels) of fg patches: large (l), medium (m), small (s) - measured by pixel diameter, with the resulting combinations being : dog images [500×375 (bg) | fg patches : 272 (l), 68 (m), 18(s)], lizard images [500×333 (bg) | fg patches : 264 (l), 66 (m), 16(s)], train images [500×334 (bg) | fg patches : 264 (l), 66 (m), 16(s)]. The resulting $3,240$ images (3 foreground/background pairs $\times$ 3 scales $\times$ 360 orientations), of which the first image from each set is shown in Figure 1 section C, do not look natural but that is not the point. By controlling the size of the foreground patch, we can rotate it without introducing orientation-specific artifacts. Figure 2 shows every 15th image from two categories of the image sets - whole images and in-place rotated objects. Figure 8 in the supplementary shows 18 rotations of the dog superimposed on the beach (blended image category). Note that we did not sample a large number of images because we are not interested in the contents of the images. We are interested in how the image encoding changes as the whole image/object/foreground is rotated, and whether these changes can be used to predict the orientation. Hence lots of orientations but few base images. The images include both indoor and outdoor scenes and both structured and less organized content.

4. Orientation Prediction Results

The goal of this experiment is to determine if the orientation of the whole image and in-place rotated object is preserved by LLaVA-One Vision and Qwen2.5-VL-7B-Instruct and similarly, if the orientation of the foreground patch is preserved by LLaVA 1.5 and 1.6. LLaVA, Qwen2.5-VL-7B-Instruct and its CLIP/SigLIP/ViT encoders are not retrained or altered in any way for this experiment. Instead, the original and rotated images, along with the textual prompt,

Figure 2. Collage of every 15th image from Sections A (a) dog (whole image) and B (a) koala (in-place rotated) of Figure 1.

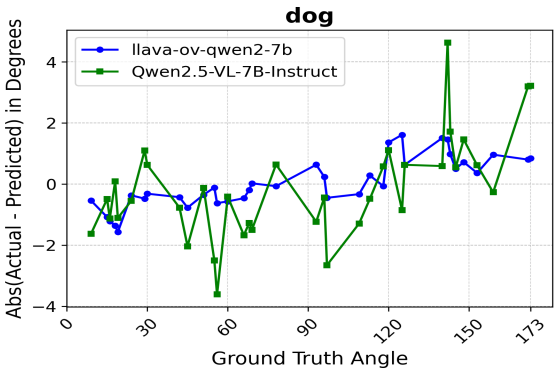


Figure 3. 2D orientation estimation performance comparison of LLaVA-OneVision and Qwen2.5-Vl-7B-Instruct on the images with the dog for 36 randomly selected images.

are presented to LLaVA-OneVision and Qwen2.5-VL-7B-Instruct or in the case of LLaVA 1.5 and 1.6, for every background image, the variations with different foreground rotations are presented along with textual prompts (see Table 1 for the prompts). For each input sample, the feature vector computed by CLIP/SigLIP/ViT is extracted from the network, flattened, and analyzed. Before flattening, the embeddings for LLaVA-OneVision are $4 \times 729 \times 1152$ values, 392×1280 or 648×1280 values for Qwen2.5-VL-7B-Instruct depending on the image size, 576×1024 values for LLAVA 1.5; for LLaVA 1.6 the embedding size depends on the image size and is $5 \times 576 \times 1024$ for the beach images, and $3 \times 576 \times 1024$ for the fish and kitchen images. The embedding vectors for each image set are divided 80:20 into training and test sets and normalized. Two independent Linear Ridge Regressors are used: one trained to predict the sine of the angle and the other to predict the cosine. Both models are provided with a predefined set of regularization parameters (α) and are trained using K-fold cross-validation to determine their respective optimal L2 regularization strengths (α). The resulting optimal α values (0.005 for both models) are then used to refit each separate Ridge Regression model on the entire training dataset to yield the final predictions for the sine and cosine of the rotation angle of the rotated image. We compare the performance of LLaVA-OneVision and Qwen2.5-VL-7B-Instruct vision encoders for the image set with the dog in Figure 3. Additional plots are presented in the supplementary material in sections 7.2 and 7.3. The Mean Absolute Error (MAE) of the predictions along with the maximum and minimum values for all image sets are presented in Table 2. The mean errors are all less then 2° except for the whole images with the fish scene and the images with the lizard foreground with the biggest foreground patch for the blended images.

4.1. Error Distributions

Surprisingly, the linear regressor was able to predict the orientation of foreground patches from the vision encoder embeddings, at least on average. This implies that LLaVA and Qwen2.5-VL-7B-312 Instruct should be able determine the 2D orientation of familiar objects, at least within familiar settings. But how easy is this information to use? Predictions with normal error distributions are easily exploitable, so how are the errors of our trained ridge regressor distributed?

To test the assumption of normality, we used a combination of statistical and visual assessment tests, as recommended by [10]. In addition to histograms, Q-Q plots, P-P plots, and Box plots, we also used the Kolmogorov-Smirnov (K-S) test. For normally distributed data, in a Q-Q plot and P-P plot, observed data would be approximate to the expected data (an approximate straight line). For a box plot, the median line would be approximately at the centre of box with symmetric whiskers. In the histogram, the graph would be approximately bell-shaped and symmetric about the mean [23]. The results of the K-S test are given in Table 3. With a low K-S statistic and a p-value greater than the significance level (alpha) of 0.05, there is no strong evidence to reject the null hypothesis of normality of the residuals [22, 23]. The visual assessment plots for LlaVA-OneVision and Qwen2.5-VL-7B-Instruct for the dog images are present in Figures 4 and 5. The results for other images and models are present in the supplementary material in Sections 7.4 and 7.5. We therefore conclude that the prediction errors are random and normally distributed, at least approximately, i.e., it is the expected stochasticity associated with a neural network.

4.2. Models used for experiments

We performed experiments with LLaVA-OneVision, LLaVA 1.5, LLaVA 1.6 (LLaVA-NeXT), and Qwen2.5-VL-7B-Instruct. LLaVA-OneVision utilizes the SigLIP vision encoder, whereas LLaVA 1.5 and 1.6 use the CLIP vision encoder. In contrast, Qwen2.5-VL-7B-Instruct uses a native Vision Transformer (ViT) backbone. While CLIP uses a softmax-based contrastive loss that requires

Two Image Version						
Whole Images						
	LLaVA-OV			Qwen Instruct		
	MAE	Max	Min	MAE	Max	Min
Dog	0.67	1.61	0.03	1.30	4.63	0.09
Lizard	0.44	1.63	0.01	2.20	6.31	0.06
Train	0.56	2.18	.01	2.43	10.01	0.29
Beach	0.74	2.67	0.03	1.63	4.98	0.03
Indoor	0.81	2.92	0.003	1.32	4.96	0.01
Fish	2.31	5.31	0.12	2.85	8.62	0.17
In-Place Rotated Object(s)						
Koala	0.36	1.47	0.01	1.01	3.66	0.06
Vase	0.48	1.76	0.003	0.90	4.18	0.01
Vase & Toaster	0.65	1.88	0.02	1.58	5.81	0.02

Single Image Version						
Blended Images						
		LLaVA 1.5			LLaVA 1.6	
		MAE	Max	Min	MAE	Max	Min
Dog	Scale 1	1.41	4.19	0.06	0.62	2.40	0.001
	Scale 2	0.89	3.20	0.0001	0.6	2.01	0.01
	Scale 3	0.67	2.52	0.02	0.52	2.2	0.03
Train	Scale 1	1.3	3.79	0.02	0.78	2.12	0.02
	Scale 2	0.9	2.69	0.003	0.53	1.72	0.0002
	Scale 3	0.72	2.12	0.008	0.42	1.2	0.005
Lizard	Scale 1	2.08	7.24	0.0003	0.87	3.15	0.04
	Scale 2	1.25	4.67	0.005	0.64	2.42	0.02
	Scale 3	1.13	3.53	0.04	0.71	1.69	0.05

Table 2. **LLaVA-OneVision**, **Qwen2.5-VL-7B-Instruct**, **LLaVA 1.5** and **LLaVA 1.6** vision encoders estimate 2D orientation to a high degree of accuracy across diverse image sets.

Two Image Version				
Whole Images				
	LLaVA-OV		Qwen Instruct	
	K-S	p-value	K-S	p-value
Dog	0.11	0.69	0.12	0.6
Lizard	0.14	0.43	0.09	0.9
Train	0.11	0.74	0.11	0.7
Beach	0.14	0.44	0.11	0.78
Indoor	0.09	0.87	0.09	0.89
Fish	0.06	0.997	0.12	0.62
In-Place Rotated Object(s)				
Koala	0.08	0.94	0.12	0.61
Vase	0.08	0.97	0.16	0.32
Vase & Toaster	0.09	0.93	0.09	0.9

Single Image Version					
Blended Images					
		LLaVA 1.5		LLaVA 1.6	
		K-S	p-value	K-S	p-value
Dog	Scale 1	0.09	0.51	0.05	0.99
	Scale 2	0.07	0.81	0.08	0.7
	Scale 3	0.1	0.46	0.12	0.37
Train	Scale 1	0.06	0.95	0.06	095
	Scale 2	0.07	0.9	0.09	0.56
	Scale 3	0.08	0.75	0.1	0.41
Lizard	Scale 1	0.06	0.96	0.08	0.74
	Scale 2	0.08	0.68	0.04	1
	Scale 3	0.06	0.95	0.08	0.69

Table 3. Low Kolmogorov–Smirnov (K-S) statistic coupled with p-value $>$ alpha (0.05) means there's no reason to reject the null hypothesis that the error distribution for **LLaVA-OneVision**, **Qwen2.5-VL-7B-Instruct**, **LLaVA 1.5** and **1.6** are purely random (Gaussian).

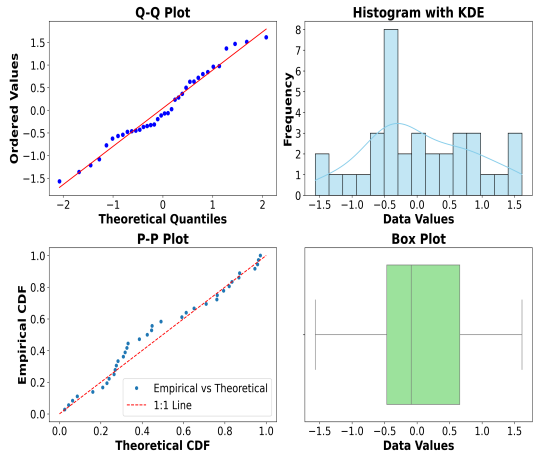


Figure 4. Statistical Analysis using visual plots for LLaVA-OneVision - results for images with dog. Together with the numerical results in Table 3 and the visual plots in this Figure, we can conclude that the residual error distribution (Table 2) is random/Gaussian.

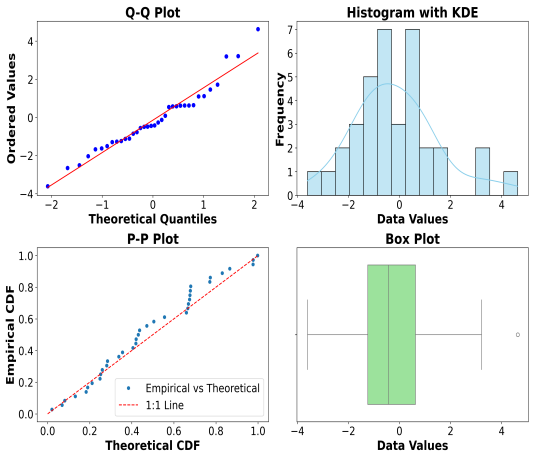


Figure 5. Statistical Analysis using visual plots for Qwen2.5-VL-7B-Instruct - results for images with dog. Together with the numerical results in Table 3 and the visual plots in this Figure, we can conclude that the residual error distribution (Table 2) is random/Gaussian.

global pairwise similarities within a batch, SigLIP introduces a sigmoid loss that operates independently on each image-text pair. This makes SigLIP more memory-efficient and performant at smaller batch sizes [40]. Despite these algorithmic differences, both SigLIP and CLIP utilize contrastive learning to achieve image-text semantic alignment [24].

4.2.1. LLaVA-OneVision

LLaVa-OneVision uses the SigLIP vision encoder and the Qwen2 LLM backbone [15]. It uses the AhnyRes-9 technique. For multi-image settings, it takes the original low-resolution image and several high-resolution patches which is a function of the source image size. The final feature is obtained by concatenating the global and the local features. In our experiments, the shape of the vision encoder output is (4,729,1152).

4.2.2. Qwen2.5-VL-7B-Instruct

The architecture of the Qwen2.5-VL series is characterized by several key technical advancements aimed at enhancing multimodal efficiency and temporal precision [1]. First, the Vision Transformer (ViT) was modified to incorporate efficient attention mechanisms, SwiGLU activations, and RMSNorm, significantly optimizing inference throughput. To support native input resolutions, the vision encoder utilizes 2D-RoPE and processes images via a dynamic resolution strategy. Second, the model extends its dynamic resolution capabilities and the spatial-temporal representation for better video understanding. Finally, these architectural improvements are supported by a massive scaling of the pre-training corpus, with the resulting visual features integrated into a Qwen2.5 language model decoder that has been specifically adapted for multimodal alignment.

4.2.3. LLaVA 1.5 vs. LLaVA 1.6

We tested 2D orientation estimation on 2 different LLaVA architectures with the CLIP vision encoder - 1.5 and 1.6. Compared with LLaVA 1.5 (llava-v1.5-13B), LLaVA 1.6 (llava-v1.6-vicuna-13B) adopts the AnyRes technique where an image is split into a grid configuration of $\{2\times2,1\times\{2,3,4\},\{2,3,4\}\times1\}$ before being fed to the language model resulting in significantly more tokens (4-5x depending on the image size for our problem) in the feature space [19].

4.3. Summary

Contrary to our initial hypothesis, we show that CLIP-style encoders (including SigLIP and ViT are trained similarly to CLIP, as detailed in Section 4.2) do preserve the orientation of familiar foreground objects with a high degree of accuracy. In fact, foreground orientations can be recovered with an MAE $< 3^\circ$ (Table 2) from LLaVA-OneVision, Qwen2.5-VL-7B-Instruct, LLaVA1.5 and LLaVA1.6 encodings. But this does not negate the findings in the literarure that MLLMs, including LLaVA, perform poorly on object orientation tasks. So if orientation information is preserved in visual embeddings, why can't LLaVA do better with orientation queries?

5. Discussion

LLaVA OneVision		Qwen2.5-VL-7B-Instruct	
Angle (°)	**Count**	**Angle (°)**	**Count**
90	149	90	142
180	28	10-15	14
0	2	10	11
45	1	180	6

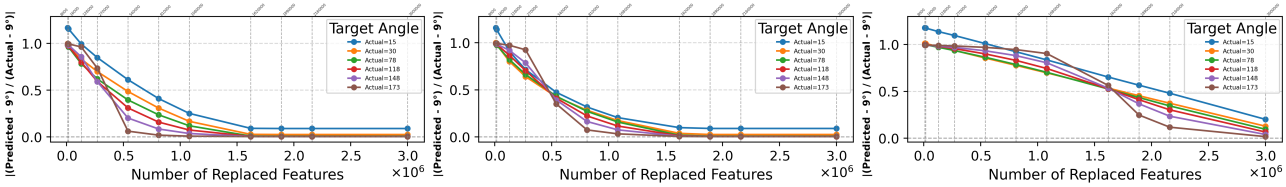
Table 4. Top LLaVA-OneVision and Qwen2.5-VL-7B-Instruct query responses for the 180 samples with the dog scene. Both models frequently respond that the 2D orientation is 90°.

Table 4 shows the query responses when estimating the 2D orientations of rotated images from the visual embeddings computed by LLaVA-OneVision and Qwen2.5-VL-7B-Instruct. Results for LLaVA 1.5 and 1.6 are given in Table 5 in the supplementary.

Since orientation information is encoded in the embedding vector and we are able to predict it accurately (Table 2), we next try to understand why the LLaVA-LLaMA models and Qwen2.5-VL-7B-Instruct perform so poorly on orientation tasks. We perform feature substitution to determine how many features are used when encoding orientation and show that the orientation information is spread diffusely across thousands of features. This may be one reason LLAVA-LLaMA and Qwen2.5-VL-7B-Instruct are unable to exploit it. We also discovered that estimates of foreground rotation depend on the background being in its standard orientation as detailed in Section 8.3 in the supplementary. Any significant rotation of the background causes the foreground estimate to become nearly random.

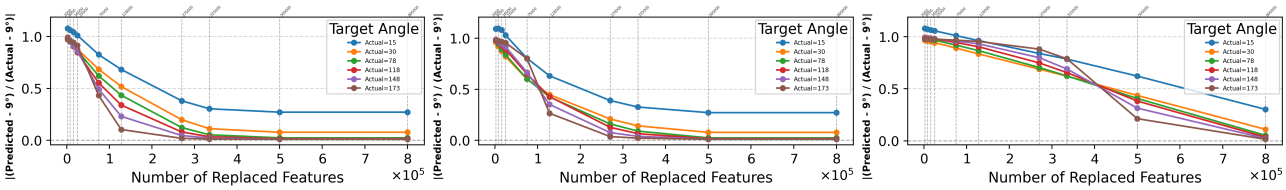
Feature Substitution Experiments We selected an anchor (in this case, the vision encoder embedding for the image where the foreground patch is rotated 9° clockwise) and replace values from the anchor's embedding vector into the embeddings for other images. The goal is to see how many embedding values have to be replaced in a non-anchor image in order for the linear predictor to believe the foreground is at 9°. We perform this experiment in 3 modes:

- Select the n embedding features with the highest weights in the linear ridge regressor, i.e., the features the predictor is more reliant on.
- Select the n embedding features with the highest absolute difference between the anchor and the target vectors, i.e. the features that change the most from anchor to target.



(a) Ordered By Model Weight (b) Ordered By Value Difference (c) Picked Randomly

Figure 6. Incremental feature substitution for LLaVA-OneVision on images with the dog scene. On the y axis, when y = 1, predicted value matches the target orientation and when y = 0, predicted value matches the anchor orientation. No matter how the features are selected (according to the magnitude of the weights in the regressor or the absolute difference between anchor and target feature values, or randomly). 540,000 features or more must be replaced to fool the predictor. (Note that the x-axis is the number of feature substitutions times 10^6.) This implies the orientation information is highly diffuse.



(a) Ordered By Model Weight (b) Ordered By Value Difference (c) Picked Randomly

Figure 7. Incremental feature substitution for Qwen2.5-VL-7B-Instruct on images with the dog scene. On the y axis, when y = 1, predicted value matches the target orientation and when y = 0, predicted value matches the anchor orientation. No matter how the features are selected (according to the magnitude of the weights in the regressor or the absolute difference between anchor and target feature values, or randomly). 128,000 features or more must be replaced to fool the predictor. (Note that the x-axis is the number of feature substitutions times 10^5.) This implies the orientation information is highly diffuse.

- Select *n* embedding features randomly (as a control).

In all 3 modes, n is varied incrementally and patch values of the target embedding are replaced with the patch values from the corresponding locations of the anchor. The results for LLaVA-OneVision and Qwen2.5-VL-7B-Instruct for the images with the dog scene are shown in Figures 6 and 7 respectively. Plots for the other images are presented in the supplementary material in sections 8.1 and 8.2. The x-axis shows the number of features from the anchor's embeddings (rotated 9°) that were substituted into the target embedding. The y-axis shows the ratio of the prediction's deviation from the anchor to the actual value's deviation from the anchor. In other words, the y axis is 1 when the predicted value matches the target orientation and 0 when it matches the anchor orientation. The more features we substitute from the anchor's embedding into the target embeddings, the closer the predicted orientation is to 9°. As expected, if mode "random selection" is considered the baseline, we see that modes "ordered by model weight" and "ordered by value difference" converge faster, with the former converging the fastest.

6. Conclusion

Many previous works have concluded that Multimodal Large Language Models (MLLMs) perform poorly on questions about orientations of objects [4, 16, 24, 25, 32–35, 45]. Some works hypothesize that it is due to the vision encoders being pre-trained on CLIP-like models (Section 2.1). This work tested and rejected this hypothesis, at least for 3 versions of LLaVA-LLaMA (OneVision, 1.5 and 1.6) and Qwen2.5-VL-7B-Instruct models. It shows that the vision encoder (CLIP/SigLIP/ViT) embeddings of these models encode the orientation of foreground patches in familiar images to within $\pm 3°$. This rejects the hypothesis, but doesn't explain why LLaVA-LLaMA and Qwen2.5-VL-7B-Instruct struggle with orientation tasks. This work can not provide a definitive causal explanation, but it does note some interesting properties of CLIP/SigLIP/ViT embeddings that may make it difficult for LLaVA-LLaMA and Qwen2.5-VL-7B-Instruct to fully exploit them. Using feature substitution, we show that orientation information is distributed across tens of thousands of features, which may make it hard for LLaVA-LLaMA and Qwen2.5-VL-7B-Instruct to learn to recover rotations. We also show that foreground orientations are sensitive to canonical background orientations, which may make them unreliable in some circumstances.

It is possible that other MLLMs may behave differently. We note, however, the orientation issues have been documented in over two dozen MLLMs (including LLaVA-LLaMA), and that many MLLMs use some version of the CLIP/SigLIP/ViT encoders. In future work, we will expand on the preliminary analysis, which showed that orientation information is diffusely encoded, which might be the reason for the poor performance of the language model.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 7

[2] Lukasz Bartoszcze, Sarthak Munshi, Bryan Sukidi, Jennifer Yen, Zejia Yang, David Williams-King, Linh Le, Kosi Asuzu, and Carsten Maple. Representation engineering for large-language models: Survey and research challenges. *arXiv preprint arXiv:2502.17601*, 2025. 2

[3] Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3836–3845, 2025. 2

[4] Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas. In *Forty-second International Conference on Machine Learning*, 2025. 1, 2, 8

[5] Kelly Cui, Nikhil Prakash, Ayush Raina, David Bau, Antonio Torralba, and Tamar Rott Shaham. The dual mechanisms of spatial reasoning in vision–language models. In *The First Workshop on Efficient Spatial Reasoning*. 2

[6] KIM Daehyun and Hyounghun Kim. Aligning vision-language models with human directional reference. 1, 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[8] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Image orientation estimation with convolutional networks. In *German conference on pattern recognition*, pages 368–378. Springer, 2015. 2

[9] Sadaf Ghaffari and Nikhil Krishnaswamy. Large language models are challenged by habitat-centered reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13047–13059, 2024. 2

[10] Farrokh Habibzadeh. Data distribution: normal or abnormal? *Journal of Korean medical science*, 39(3), 2024. 5

[11] Ngoc Dung Huynh, Yasser Dahou, Phuc H Le-Khac, Wamiq Reyaz Para, Ankit Singh, and Sanath Narayan. Vision-language models can't see the obvious. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24159–24169, 2025. 2

[12] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2025. 1, 2

[13] Ji Hyeok Jung, Eun Tae Kim, Seoyeon Kim, Joo Ho Lee, Bumsoo Kim, and Buru Chang. Isright'right? enhancing object orientation understanding in multimodal large language models through egocentric instruction tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14257–14267, 2025. 2

[14] Ryo Kamoi, Yusen Zhang, Sarkar Snigdha Sarathi Das, Ranran Haoran Zhang, and Rui Zhang. Visonlyqa: Large vision language models still struggle with visual perception of geometric information. *arXiv preprint arXiv:2412.00947*, 2024. 1

[15] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1, 7

[16] Shijie Lian, Changti Wu, Laurence Tianruo Yang, Hang Yuan, Bin Yu, Lei Zhang, and Kai Chen. Euclid's gift: Enhancing spatial perception and reasoning in vision-language models via geometric surrogate tasks. *arXiv preprint arXiv:2509.24473*, 2025. 1, 8

[17] Disheng Liu, Tuo Liang, Zhe Hu, Jierui Peng, Yiren Lu, Yi Xu, Yun Fu, and Yu Yin. Spatial intelligence in vision-language models: A comprehensive survey. 2026. 2

[18] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 1

[19] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 1, 7

[20] Dongchen Lu, Dongmei Li, Yali Li, and Shengjin Wang. Oskdet: Orientation-sensitive keypoint localization for rotated object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1182–1192, 2022. 2

[21] Xueqi Ma, Shuo Yang, Yanbei Jiang, Shu Liu, Zhenzhen Liu, Jiayang Ao, Xingjun Ma, Sarah Monazam Erfani, and James Bailey. Attention in space: Functional roles of vlm heads for spatial reasoning. *arXiv preprint arXiv:2603.20662*, 2026. 1, 2

[22] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951. 5

[23] Prabhaker Mishra, Chandra M Pandey, Uttam Singh, Anshul Gupta, Chinmoy Sahu, and Amit Keshri. Descriptive statistics and normality tests for statistical data. *Annals of cardiac anaesthesia*, 22(1):67–72, 2019. 5

[24] Keanu Nichols, Nazia Tasnim, Yuting Yan, Nicholas Ikechukwu, Elva Zou, Deepti Ghadiyaram, and Bryan A Plummer. Right side up? disentangling orientation understanding in mllms with fine-grained multi-axis perception tasks. *arXiv preprint arXiv:2505.21649*, 2025. 1, 2, 3, 7, 8

[25] Tianyi Niu, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. Rotbench: Evaluating multimodal large language models on identifying image rotation. *arXiv preprint arXiv:2508.13968*, 2025. 1, 2, 8

[26] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial represen-

tations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 2

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 3

[29] Fatemeh Shiri, Xiao-Yu Guo, Mona Far, Xin Yu, Reza Haf, and Yuan-Fang Li. An empirical analysis on spatial reasoning capabilities of large multimodal models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21440–21455, 2024. 2

[30] Jie Sun, Wengang Zhou, and Houqiang Li. Orientation estimation network. In *International Conference on Image and Graphics*, pages 151–162. Springer, 2017. 2

[31] Bowei Tian, Xuntao Lyu, Meng Liu, Hongyi Wang, and Ang Li. Why representation engineering works: A theoretical and empirical study in vision-language models. *arXiv preprint arXiv:2503.22720*, 2025. 2

[32] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024. 1, 2, 8

[33] Siting Wang, Minnan Pei, Luoyang Sun, Cheng Deng, Kun Shao, Zheng Tian, Haifeng Zhang, and Jun Wang. Spatialviz-bench: An mllm benchmark for spatial visualization. *arXiv preprint arXiv:2507.07610*, 2025.

[34] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025.

[35] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025. 1, 2, 8

[36] Heeji Yoon, Jaewoo Jung, Junwan Kim, Hyungyu Choi, Heeseong Shin, Sangbeom Lim, Honggyu An, Chaehyun Kim, Jisang Han, Donghyun Kim, et al. Visual representation alignment for multimodal large language models. *arXiv preprint arXiv:2509.07979*, 2025. 2

[37] Songsong Yu, Yuxin Chen, Hao Ju, Lianjie Jia, Fuxi Zhang, Shaofei Huang, Yuhan Wu, Rundi Cui, Binghao Ran, Zaibin Zhang, et al. How far are vlms from visual spatial intelligence? a benchmark-driven perspective. *arXiv preprint arXiv:2509.18905*, 2025. 1, 2

[38] Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025. 2

[39] Jessica Yung, Rob Romijnders, Alexander Kolesnikov, Lucas Beyer, Josip Djolonga, Neil Houlsby, Sylvain Gelly, Mario Lucic, and Xiaohua Zhai. Si-score: An image dataset for fine-grained analysis of robustness to object location, rotation and size. *arXiv preprint arXiv:2104.04191*, 2021. 3

[40] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 1, 7

[41] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018. 2

[42] Wanyue Zhang, Yibin Huang, Yangbin Xu, JingJing Huang, Helu Zhi, Shuo Ren, Wang Xu, and Jiajun Zhang. Why do mllms struggle with spatial understanding? a systematic analysis from data to architecture. *arXiv preprint arXiv:2509.02359*, 2025. 1

[43] Wenqi Zhang, Mengna Wang, Gangao Liu, Xu Huixin, Yiwei Jiang, Yongliang Shen, Guiyang Hou, Zhe Zheng, Hang Zhang, Xin Li, et al. Embodied-reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks. *arXiv preprint arXiv:2503.21696*, 2025. 2

[44] Xu Zheng, Zihao Dongfang, Lutao Jiang, Boyuan Zheng, Yulong Guo, Zhenquan Zhang, Giuliano Albanese, Runyi Yang, Mengjiao Ma, Zixin Zhang, et al. Multimodal spatial reasoning in the large model era: A survey and benchmarks. *arXiv preprint arXiv:2510.25760*, 2025. 1, 2

[45] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 1, 8